Opérateur DEFI_FONCTION

1 But

Définir une fonction réelle ou complexe d'une variable réelle. Cet opérateur permet de définir, par exemple, des caractéristiques matériaux fonction de la température, ou des conditions aux limites qui dépendent d'une variable d'espace ou du temps.

Le concept produit par cet opérateur est de type `fonction`.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Syntaxe

```
f  [fonction]   =  DEFI_FONCTION (

                   ♦  NOM_PARA  =  np,

                   ◊  NOM_RESU  =  /  'TOUTRESU',              [DEFAUT]
                                   /  nr,                      [K8]

                   ♦  /  VALE         =  lv,                   [l_R]
                      /  VALE_C       =  lv,                   [l_C]
                      /  VALE_PARA    =  la,                   [listr8]
                         ♦  VALE_FONC    =  lo,                [listr8]
                      /  ABSCISSE     =  labs,                 [l_R]
                         ♦  ORDONNEE     =  lord,              [l_R]
                      /  NOEUD_PARA   =  lno,                  [l_noeud]
                         ♦  MAILLAGE     =  ma,                [maillage]
                         ♦  VALE_Y       =  ly,                [l_R]

                   ◊  PROL_DROITE  =  /  'CONSTANT',
                                      /  'LINEAIRE',
                                      /  'EXCLU',              [DEFAUT]

                   ◊  PROL_GAUCHE  =  /  'CONSTANT',
                                      /  'LINEAIRE',
                                      /  'EXCLU',              [DEFAUT]

                   ◊  INTERPOL  =  |  'LIN',                   [DEFAUT]
                                   |  'LOG',
                                   |  'NON',

                   ◊  INFO   =     /  1,                       [DEFAUT]
                                   /  2,

                   ◊  VERIF  =  |  'CROISSANT',                [DEFAUT]
                                |  'NON',

                   ◊  TITRE  =  ti,                            [l_Kn]

                                   )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Opérandes

3.1 Opérande NOM_PARA

♦ `NOM_PARA = np`

Désigne le nom du paramètre (variable ou abscisse) de la fonction.
Les valeurs possibles pour `np` sont :

`'ABSC', 'AMOR', 'DRX', 'DRY', 'DRZ', 'DSP', 'DX', 'DY', 'DZ', 'ENDO', 'EPAIS', 'EPSI', 'FREQ', 'HYDR', 'INST', 'META', 'NEUT1', 'NEUT2', 'NORM', 'PAD', 'PCAP', 'PGAZ', 'PLIQ', 'PORO', 'PULS', 'PVAP', 'SAT', 'SECH', 'SIGM', 'TEMP', 'TSEC', 'VITE', 'X', 'Y', 'Z'`

3.2 Opérande NOM_RESU

◊ `NOM_RESU = nr`

Désigne le nom du résultat (8 caractères). La fonction ainsi créée est `nr` = f(`np`).

Remarque :

Certaines commandes (`CALC_FONCTION` , `DEFI_MATERIAU` …) vérifient la cohérence des noms du paramètre et du résultat en fonction de leur contexte. Par exemple, on attend une courbe de traction définie par une fonction dont `NOM_PARA='EPSI'` et `NOM_RESU='SIGM'`.

3.3 Opérande VALE

/ `VALE = lv`

`lv` est la liste de valeurs $(x_1, y_1, …, x_n, y_n)$ avec dans l'ordre :

- x_1, y_1 (la première valeur du paramètre et la valeur correspondante du résultat),
- ... ,
- x_n, y_n (la dernière valeur du paramètre et la valeur correspondante du résultat).

Remarque :

La liste `lv` de valeurs doit être décrite dans l'ordre des abscisses (x) croissantes.

3.4 Opérande VALE_C

/ `VALE_C = lv`

`lv` est la liste des valeurs $(x, y, z, …, x_n, y_n, z_n)$ avec :

- x_i valeurs du paramètre
- ... ,
- y_i, z_i la partie réelle et la partie imaginaire de la fonction complexe pour ce paramètre.

3.5 Opérandes ABSCISSE / ORDONNEE

/ `ABSCISSE = labs`
/ `ORDONNEE = lord`

On fournit les valeurs des abscisses et des ordonnées de la fonction séparément sous la forme de listes de valeurs réelles $(x_1, x_2, …, x_n)$ pour `ABSCISSE` et $(y_1, y_2, …, y_n)$ pour `ORDONNEE`. Les deux listes doivent avoir le même cardinal.

3.6 Opérande VALE_PARA / VALE_FONC

/ `VALE_PARA = la`

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

/ `VALE_FONC = lo`

Même fonctionnement que `ABSCISSE`, `ORDONNEE` sauf que les listes sont fournies sous la forme de concept `listr8` produit par `DEFI_LIST_REEL` [U4.34.01].
`VALE_PARA` et `VALE_FONC` doivent être des cardinaux identiques sinon la commande s'arrête en erreur.

3.7 Opérande `NOEUD_PARA`

/ `NOEUD_PARA = lno`

`lno` liste de nœuds permettant de définir les valeurs des abscisses de la fonction à définir.
Les abscisses seront égales aux abscisses curvilignes des nœuds sur la courbe qu'ils définissent.

3.8 Opérandes `PROL_DROITE` et `PROL_GAUCHE`

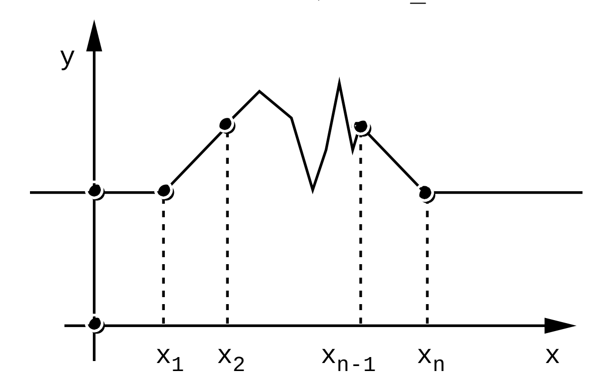
◊ `PROL_DROITE` et `PROL_GAUCHE =`

Définissent le type de prolongement à droite (à gauche) du domaine de définition de la variable :

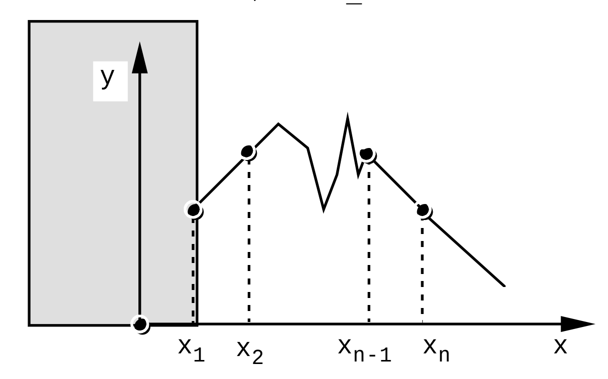
- `'CONSTANT'` pour un prolongement avec la dernière (ou première) valeur de la fonction,
- `'LINEAIRE'` pour un prolongement le long du premier segment défini (`PROL_GAUCHE`) ou du dernier segment défini (`PROL_DROITE`),
- `'EXCLU'` si l'extrapolation des valeurs en dehors du domaine de définition du paramètre est interdite (dans ce cas si un calcul demande une valeur de la fonction hors du domaine de définition, le code s'arrêtera en erreur fatale).

Par exemple :

- `PROL_DROITE = 'CONSTANT' , PROL_GAUCHE = 'CONSTANT'`



- `PROL_DROITE = 'LINEAIRE', PROL_GAUCHE = 'EXCLU'`



Remarque :

Le type de prolongement et d'interpolation sont indépendants l'un de l'autre.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3.9 Opérande INTERPOL

◊ INTERPOL =

Type d'interpolation de la fonction entre les valeurs du domaine de définition de la fonction : un type pour l'interpolation du paramètre et un pour l'interpolation de la fonction. Ceci est obtenu en fournissant une liste de textes parmi :

INTERPOL = ('LIN', 'LOG')

'LIN' : linéaire,

'LOG' : logarithmique,

'NON' : on n'interpole pas (et donc le programme s'arrêtera si l'on demande la valeur de la fonction pour une valeur du paramètre où elle n'a pas été définie).

Remarque :

Si une seule valeur est précisée, elle est prise en compte à la fois par l'interpolation du paramètre et de la fonction. INTERPOL = 'LOG' équivaut à ('LOG', 'LOG').

3.10 Opérande INFO

◊ INFO = Précise les options d'impression sur le fichier MESSAGE.

1 : pas d'impression (option par défaut)

2 : impression des paramètres plus la liste des 10 premières valeurs dans l'ordre croissant du paramètre

3.11 Opérande VERIF

◊ VERIF =

L'opérateur DEFI_FONCTION vérifie que les valeurs des abscisses sont strictement croissantes. Si ce n'est pas le cas, une erreur est déclenchée. Ceci est le comportement par défaut, VERIF vaut 'CROISSANT'.
L'utilisateur a la possibilité de ne pas faire cette vérification en indiquant VERIF='NON'. Dans ce cas, la fonction est réordonnée par abscisses croissantes. Une alarme est émise si les abscisses de la fonction n'étaient pas croissantes.

3.12 Opérande TITRE

◊ TITRE = ti

Titre attaché au concept produit par cet opérateur [U4.03.01].

3.13 Opérandes MAILLAGE et VALE_Y

Il faut renseigner ces deux mots-clés si on définit la fonction à partir de NOEUD_PARA.

MAILLAGE = ma

Nom du maillage associé à la liste de nœud lno.

VALE_Y = lv

Liste des valeurs des ordonnées de la fonction à définir.

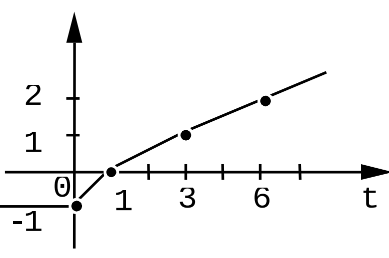
Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4 Définition d'une fonction dépendant du temps

4.1 Fonction et variables sont entrées sous forme de réels

Définition d'un fonction (linéaire par morceaux) dépend du temps (paramètre `INST`).

```
EX_1 = DEFI_FONCTION  (
                NOM_PARA='INST',
                VALE = (  0. ,-1.,
                          1. , 0.,
                          3. , 1.,
                          6. , 2., ),
             PROL_GAUCHE='CONSTANT',
             PROL_DROITE='LINEAIRE',
                   )
```



4.2 Fonction et variables sont entrées sous forme de concepts `listr8`

Il est possible de définir cette fonction à l'aide de concepts de type `listr8` créés par l'intermédiaire de l'opérateur `DEFI_LIST_REEL` [U4.34.01] :

```
ABSCISSE = DEFI_LIST_REEL  (  DEBUT = 0. ,
                INTERVALLE =  ( _F ( JUSQU_A = 1., NOMBRE = 1,),
                                _F ( JUSQU_A = 3., NOMBRE = 1,),
                                _F ( JUSQU_A = 6., NOMBRE = 1,),)
                           )
ORDONNEE = DEFI_LIST_REEL  (  DEBUT = -1. ,
                INTERVALLE =  ( _F (JUSQU_A = 0., NOMBRE = 1,),
                                _F (JUSQU_A = 1., NOMBRE = 1,),
                                _F (JUSQU_A = 2., NOMBRE = 1,),)
                           )
EX_2 = DEFI_FONCTION   (  NOM_PARA    = 'INST',
                          VALE_PARA   = ABSCISSE,
                          VALE_FONC   = ORDONNEE,
                          PROL_DROITE = 'CONSTANT',
                          PROL_GAUCHE = 'LINEAIRE',
                       )
```

Remarque :

Cet exemple est évidemment bien compliqué pour définir la fonction proposée.
Nous n'avons voulu que mettre en évidence le principe d'utilisation de la possibilité offerte.
Celle-ci devient intéressante lorsque l'on utilise des fonctions définies en un grand nombre de points.
Une autre raison d'utiliser la définition par `DEFI_LIST_REEL` est lorsque les listes sont nécessaires comme argument pour un autre opérateur : (liste des instants d'un calcul évolutif `THER_LINEAIRE` , `DYNA_LINE_TRAN` , …) , ceci évite alors la duplication d'informations.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)